REMARKS

Applicant thanks the Examiner for total consideration given the present application. Claims 1-19 were pending prior to the final Office Action. Claims 2, 5, 6, 13, 15, 17, and 19 have been canceled through this Reply. Therefore, claims 1, 3-4, 7-12, 14, 16, and 18 are currently pending of which claims 1, 9, 16, and 18 are independent. Claims 1, 7, 14, 16, and 18 have been amended through this Reply. Applicant respectfully requests reconsideration of the rejected claims in light of the amendment and remarks presented herein, and earnestly seeks timely allowance of all pending claims.

Claim Rejection - 35 U.S.C. § 101

Claims 1 and 9 stand rejected under 35 U.S.C. § 101 as allegedly being directed to non-statutory subject matter. Particularly, the Examiner alleges that independent claims 1 and 9 are not statutory since no requisite functionality is present to satisfy the practical application requirement. (*See page 3, lines 6-8 of the final Office Action.*) It is respectfully submitted that Examiner's such allegation is totally unfounded. Previously presented claims 1 and 9 included the requisite functionality, such as implementing the constructs described by metadata, describing attributes of the model element class, accessing the attribute value stored in the model element class.

Although Applicant respectfully disagrees with the Examiner's contention that independent claims 1 and 9 are directed to non-statutory subject matter, these claims have been amended to recite, *inter alia*, "configured to implement the constructs described by metadata . . . configured to describe attributes of the model element class . . . configured to access the attribute value stored in the model element class . . ." merely to expedite prosecution.

Accordingly, it is respectfully requested to withdraw this rejection.

MKM/AMI/bms

35 U.S.C. § 102 REJECTION - Brumme

Claims 1, 4, 8-10, and 12 stand finally rejected under 35 U.S.C. § 102(b) as allegedly being anticipated by Brumme et al. (U.S. Patent 6,134,559)[hereinafter "Brumme"]. Applicant respectfully traverses this rejection.

For a Section 102 rejection to be proper, the cited reference must teach or suggest each and every claimed element. *See M.P.E.P. 2131; M.P.E.P. 706.02.* Thus, if the cited reference fails to teach or suggest one or more elements, then the rejection is improper and must be withdrawn.

In this instance, Brumme fails to teach or suggest each and every claimed element. For example, independent claim 1 recites, *inter alia,*

> "a model element class configured to implement the constructs described by metadata; *the model element class storing an attribute value in a private member field of the model element class in the same memory block as a declared nested handler class,* the nested handler class being a subclass of a generic handler class and inherits base functionality from the generic handler class, wherein *the nested handler class has public access to an enclosing element's private members stored in the private member field of the model element class . . .*
>
> a model element field handler object configured to access the attribute value stored in the model element class, *wherein the model element field handler object comprises a typed model element field handler subclass defining a get value function wherein the get value function is configured to access the model element class and return the attribute value directly upon request;* and
>
> *wherein the storage of the attribute value is separate from handling of the attribute value." Emphasis added.*

It is respectfully submitted that Brumme fails to teach or suggest the above-identified claim features of independent claim 1. Brumme merely discloses a conventional method of integrating foreign objects and different data sources into a single integrated object oriented environment. Particularly, Brumme is directed to provide an integrated object oriented system that includes an integrated type system for supporting *a superset of features* from a plurality of *foreign object systems* wherein the foreign object systems comprises a plurality of foreign objects defined by foreign type systems that are different from the integrated type system.

Burmme further discloses that the foreign objects are being converted into uniform object model objects defined by the integrated type system. (See col. 4, lines 2-28.) The entire reference is silent on whether *a model element class* is configured to implement the constructs described by metadata and stores an attribute value in a private member field of the model element class *in the same memory block* as a declared nested handler class wherein the nested handler class has public access to an enclosing element's private members stored in the private member field of the model element class. In addition, Brumme fails to teach or suggest *a model element field handler object* that is configured *to access the attribute value stored in the model element class*, wherein the model element field handler object comprises *a typed model element field handler subclass* defining a get value function *wherein the get value function is configured to access the model element class and return the attribute value directly upon request.* *Emphasis added.*

The Examiner relies on col. 3, lines 17-30, col. 22, lines 32-63, and col. 14, lines 51-65 of Brumme to allege that Brumme teaches, "*wherein the storage of the attribute value is separate from handling of the attribute value.*" Applicant respectfully submits that the Examiner's interpretation of the relied upon section of Brumme is clearly erroneous. Neither the cited portions nor any other portions of Brumme teach or suggest that the storage of the attribute value is separate from handling of the attribute value. For example, Brumme discloses at col. 3, lines 17-30 as follows:

> One type of database used in software systems is a relational database. In general, a relational database stores information in the form of tables. Typically, relational databases include software that permit a user, through a query language, to visualize different portions and combinations of these tables. Object oriented software systems, which include encapsulated data and methods, do not have a direct counterpart with the relational model. For example, an object may include several attributes, specified by the class type, that define the state of the object at a particular time. *To store the attributes of an object*, an object oriented concept, in a relational database, *a mapping of the attributes of the object to columns and rows in one or more tables is required.*

After carefully reviewing the above section and other cited portions indicated by the Examiner, Applicant finds no teaching or suggestion as to whether the storage of the attributes of

an object is separate from handling of attributes. The Examiner's mere assertion that "attributes are handled by handlers" (see page 4, section 3.d. of the final Office Action) does not support the allegation that Brumme teaches or suggests that the storage of the attribute value is separate from the handling of the attribute value.

Therefore, for at least these reasons, independent claim 1 is distinguishable from Brumme. Independent claim 9 also recites subject matter similar to claim 1. Therefore, at least for the reasons stated with respect to claim 1, claim 9 is also distinguishable from Brumme Further, independent claim 9 recites, *inter alia*,

> "*a model element field handler object* configured to access the attribute value stored in the model element class, wherein *the model element field handler object comprises a typed model element field handler subclass defining a set value function* for setting the attribute value;
> wherein the nested handler class is configured to *directly access data in the model element class as the nested handler class has public access to the private members of the model element class . . .*"

Applicant respectfully submits that Brumme fails to teach or suggest the above-identified claim features of independent claim 9. As demonstrated above, Brumme merely discloses a step for converting *foreign objects* into *uniform object model objects* defined by an integrated type system. Brumme is silent on whether there is *a model element field handler object* as recited in independent claim 9.

At least for the reasons stated above, independent claims 1 and 9 are distinguishable from Brumme. Claims 4 and 8 depend from claim 1 and claims 10 and 12 depend from claim 9. Therefore, for at least the reasons stated claims 1 and 9, claims 4, 8, 10, and 12 are also distinguishable from Brumme.

Accordingly, Applicant respectfully requests that the rejection of claims 1, 4, 8-10, and 12, based on Brumme, be withdrawn.

Please note that the amendment to independent claim 1 includes, *inter alia*, subject matters from currently cancelled claims 2, 5, and 6 and amendment to independent claim 9 includes, *inter alia*, subject matters from currently cancelled claims 13 and 15. The Examiner relies on Mathews et al. (US 2003/0163479) to reject claim 2 and Coad et al. (US 6,851,105) to reject claims 5, 6, 13, and 15. Although the Applicant does not necessarily agree with the Examiner's assertion that Mathews teaches that the attribute value is stored in private member field of the model element class, claims 1 and 9 have been further amended to clarify that *the model element class storing an attribute value in a private member field of the model element class in the same memory block as a declared nested handler class.* It is respectfully submitted that Mathews fails to teach or suggest the above-identified claim feature.

Further, Applicant submits that neither the cited portions nor any other portions of Coad teach or suggest a model element field handler object including *a typed model element field handler subclass* defining a get value function *wherein the get value function is configured to access the model element class and return the attribute value directly upon request* as recited in independent claim 1. In addition, Coad is silent on whether the model element field handler object comprises *a typed model element field handler subclass defining a set value function for setting the attribute value,* wherein *the nested handler class is configured to directly access data in the model element class as the nested handler class has public access to the private members of the model element class* as recited in independent claim 9. The Examiner's mere allegation, which Applicant respectfully disagrees, that get and set functions are common generic functions used to get and set values for attributes in classes, subclasses, or just with regular functions, does not necessarily mean that the get value function is configured to access the model element class and return the attribute value directly upon request nor does it mean that the typed model element field handler subclass defining a set value function for setting the attribute value, wherein the nested handler class is configured to directly access data in the model element class as the nested handler class has public access to the private members of the model element class.

Accordingly, it is respectfully submitted that claims 1, 4, 8-10, and 12 are allowable over Brumme, Mathews, and Coad.

11   MKM/AMI/bms

Claim 2 is rejected under 35 U.S.C. § 103(a) as allegedly being unpatentable over Brumme in view of Mathews. Claim 2 has been cancelled through this reply rendering this rejection moot.

Claims 3, 5-7, 11, and 13-15 are rejected under 35 U.S.C. § 103(a) as allegedly being unpatentable over Brumme in view of Coad. Claims 5, 6, 13, and 15 have been cancelled through this reply rendering the rejection of theses claims moot. Claims 3, 7, 11, and 14 are at least allowable by virtue of their dependency on corresponding independent claim.

Claims 16-19 are rejected under 35 U.S.C. § 103(a) as allegedly being unpatentable over Brumme in view of Brogden et al. (Java 2 Progammer Exam Cram 2)[hereinafter "Brogden"]. Applicant respectfully traverses this rejection. It has been demonstrated above that Brumme fails to teach or suggest, *"wherein the storage of the attribute value is separate from handling of the attribute value"* as recited in claims 16 and 18. Further, claim 16 has been amended to include, *inter alia*, *"wherein the nested handler class has public access to an enclosing element's private members stored in the private member field of the model element class; . . . wherein the get value function is configured to access the model element class and to return the attribute value directly upon request".* Claim 18 has also been amended to include, *inter alia*, "wherein *the nested handler class is configured to directly access data in the model element class as the nested handler class has public access to the private members of the model element class".*

Brogden merely discloses why and how to create a nested class. Although Brogden discloses that nested classes have unlimited access, nowhere does Brogden teach or suggest that nested handler class has public access to an enclosing element's private members *stored in the private member filed* of the *model element class* as recited in claims 16 and 18.

At least for the reasons stated above, it is respectfully submitted that claims 16 and 18 are allowable over Brumme and Borgden. Claims 17 and 19 have been cancelled through this reply thereby rendering the rejection of these claims moot.

CONCLUSION

In view of the above remarks, it is believed that all pending claims are allowable.

Should there be any outstanding matters that need to be resolved in the present application, the Examiner is respectfully requested to contact Ali M. Imam Reg. No. 58,755 at the telephone number of the undersigned below, to conduct an interview in an effort to expedite prosecution in connection with the present application.

If necessary, the Commissioner is hereby authorized in this, concurrent, and future replies to charge payment or credit any overpayment to Deposit Account No. 02-2448 for any additional fees required under 37.C.F.R. §§1.16 or 1.14; particularly, extension of time fees.

Dated: August 20, 2007   Respectfully submitted,

                                                 By _____ # 58,755
                                                 Michael K. Mutter
                                                 Registration No.: 29,680
                                                 BIRCH, STEWART, KOLASCH & BIRCH, LLP
                                                 8110 Gatehouse Road
                                                 Suite 100 East
                                                 P.O. Box 747
                                                 Falls Church, Virginia 22040-0747
                                                 (703) 205-8000
                                                 Attorney for Applicant

                                                            MKM/AMI/bms